LARAVEL HINTS:

php -S localhost:8000 -t public/

Laravel install

**composer create-project --prefer-dist laravel/laravel blog**

**Default Length Error in laravel:**

**App->providers->AppServiceProvider.php**

**Add Following line**

**use Illuminate\Support\Facades\Schema;**

**public function boot()**

**{**

**Schema::defaultStringLength(191);**

**}**

**Creating a Controller:**

**php artisan make:controller PostController**

**Creating a new Model AND Schema creation**

php artisan make:model Post -m

This command will create a new Post model and because we added -m or --migration flag, it will create a migration for it as well. This command will create two files at app/Post.php and database/migrations/create\_posts\_table.php. Create posts table relationship with the user by opening app/User.php and defining the following relation:

**Model creation:**

**php artisan make:model Category**

**Model include in CategoryController:**

**use App\Category;**

**Form save and redirect in controller:**

**<?php**

**namespace App\Http\Controllers;**

**use Illuminate\Http\Request;**

**use App\Category;**

**class CategoryController extends Controller**

**{**

**public function category(){**

**return view("categories.category");**

**}**

**public function addCategory(Request $request){**

**$this->validate($request,[**

**'category'=>'required'**

**]);**

**$category = new Category;**

**//Here $category->category -db column name**

**//$request->input('category');-form column name**

**//redirect to AddForm page**

**$category->category = $request->input('category');**

**$category->save();**

**return redirect('/category')->with('respose','Category Added Successfully');**

**}**

**}**

**Success Messsage in view page Controller with respose:**

**@if(count($errors) > 0 )**

**@foreach($error->all() as $error)**

**<div class="alert alert-danger">{{$error}}</div>**

**@endforeach**

**@endif**

**@if(session()->has('respose'))**

**<div class="alert alert-success">**

**{{ session()->get('respose') }}**

**</div>**

**@endif**

**Forloop:**

**<ul class="list-group">**

**@if(count($categories) > 0 )**

**@foreach($categories->all() as $category)**

**<li class="list-group-item"><a href='{{url("category/{$category->id}")}}'>{{$category->category}}</a></li>**

**@endforeach**

**@else**

**<p>Nocategory Found</p>**

**@endif**

**</ul>**

**Auth check in home blade:**

@if (Auth::check())

//show logged in navbar

@else

//show logged out navbar

@endif

**Normal Table creation:**

php artisan make:migration create\_links\_table --create=links

**Table migration**

php artisan migrate

php artisan migrate:refresh

## Creating Database Seeds

Database seeds are a way of creating fake data and adding it to your database. It helps you get fake data quickly than creating it manually. Let’s create a seed for users and posts table. To do this run both of the following commands one by one:

**php artisan make:seeder UsersTableSeeder**

**php artisan make:seeder PostsTableSeeder**

<?php

use Illuminate\Database\Seeder;

use App\Article; class ArticlesTableSeeder extends Seeder

{

/\*\*

\* Run the database seeds.

\*

\* @return void

\*/

public function run()

{

// Let's truncate our existing records to start from scratch.

Article::truncate();

$faker = \Faker\Factory::create();

// And now, let's create a few articles in our database:

for ($i = 0; $i < 50; $i++) {

Article::create([

'title' => $faker->sentence,

'body' => $faker->paragraph,

]);

}

}

} }

**php artisan db:seed**

<https://www.youtube.com/watch?v=nXq4ndJvrhg>

Laravel tutorial Blog creation:

Home page Link add in file:

app.blade.php

site name change.

**{{ config('app.name', 'Laravel') }}**

To

Blog

Url add:

**<li><a href="{{ url('/home') }}">Home</a></li>**

**<li><a href="{{ url('/post') }}">Add Post</a></li>**

1.create 3 folders.

View->categories->category.php

View->profiles->profile.php

View->posts->post.php

**Copy** [**home.blade.php**](http://home.blade.php) **and paste Each files like**

category.php,

profile.php,

post.php

**web.php Routes:**

**Route::get('/post', 'PostController@post');**

**In post controller:**

**public function post(){**

**return view("posts.post");**

**}**

**Create new table schema:**

**php artisan make:migration create\_posts\_table --create=posts**

**php artisan migrate**

All table inserted

Create category form:

Go login.blade.php copy form and paste in category page

**How to get post valuein laravel:**

Web.php

Here addCategory is Form action

**Route::post('/addCategory', 'CategoryController@addCategory');**

**CategoryController.php:**

**public function addCategory(Request $request){**

**return $request->input('category');**

**}**

**Server side validation:**

**public function addCategory(Request $request){**

**$this->validate($request,[**

**'category'=>'required'**

**]);**

**return "validation passed";**

**}**

**How to get user id in controller:**

**use Auth;//Packages**

**$profile->user\_id = Auth::user();**

**Debug user:**

**$profile->user\_id = Auth::user();**

**return Auth::user();**

**exit;**

**Output:**

**{"id":14,"name":"stalin","email":"stalingalaxy@gmail.com","email\_verified\_at":null,"created\_at":"2019-03-29 17:00:08","updated\_at":"2019-03-29 17:00:08"}**

**$profile->user\_id = Auth::user()->id;//14**

**File upload Packages:**

**use Illuminate\Support\Facades\Input;**

**use Illuminate\Support\Facades\File;**

**use Illuminate\Support\Facades\URL;**

**Upload coding:**

**Create a folder following path**

**public/uploads**

**if(Input::hasFile('profile\_pic')){**

**$file = Input::file('profile\_pic');**

**$file->move(public\_path().'/uploads/',$file->getClientOriginalName());**

**$url = URL::to("/"). "/uploads/".$file->getClientOriginalName();**

**}**

Api tutorial:

<https://www.toptal.com/laravel/restful-laravel-api-tutorial>

1. php artisan make:model Article -m

php artisan make:controller PostController --resource

3.php artisan make:resource Article

REST API:

CRPF Token Error:

App->http->kernel.php

Comment crpf

// App\Http\Middleware\VerifyCsrfToken::class,

In mode:

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Post extends Model

{

//very important in rest api add,edit,delete

protected $fillable = ['title', 'body'];

}

POST:

<http://localhost/rd/rest/public/articles>

body-> form-data

'title': stalin

'body':Thomas

{

"title": "Stalin",

"body": "Thomas",

"updated\_at": "2019-03-31 09:05:35",

"created\_at": "2019-03-31 09:05:35",

"id": 33

}

PUT:

<http://localhost/rd/rest/public/articles/31>

body->raw->application/json

{

"title":"stalin",

"body":"welcome Chennai"

}

DELETE:

<http://localhost/rd/rest/public/articles/31>

OK

SQLITE:

touch database/database.sqlite

go to .envi file.

remove

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=lara

DB\_USERNAME=root

DB\_PASSWORD=

Instead

DB\_CONNECTION=sqlite

Go to databse.php

'sqlite' => [

'driver' => 'sqlite',

'database' => env('DB\_DATABASE', database\_path(database.sqlite')),

'prefix' => '',

],

Basic Operations:

select from rooms where id=1;

$id= 1;

**Room::find($id);**

select \* from rooms where name='Lovely Hotel';

**Room:: where('name','Lovely Hotel')->get();**

Like:

**Room:: where where('name','like','%Lovely%')->get();**

**select \* from accommodations where name ='Lovely Hotel' and**

**description like '%Pittsburgh%'**

**MyCompany\Accommodation::where('name','Lovely Hotel')-**

**>where('city','like','%Pittsburgh%')->get();**

MyCompany\Accommodation::where('name','Lovely Hotel')-

>orWhere('description','like','%Pittsburgh%')->get();

**select \* from accommodations where name ='Lovely Hotel' or**

**description like '%Pittsburgh%'**

**Stripe Account for Laravel’s cashier packages:**

**Email:**

[**stalingalaxy@gmail.com**](mailto:stalingalaxy@gmail.com)

**password:**

**dilma123**

**Publishable Key:**

**pk\_test\_t9JIYDUJDlhjoUBwKWyH5sr900NFtlF5bj**

**secret key:**

**sk\_test\_xlJ1nlMXtUESup7CzmDk3Rw70079C3Hfuj**

**Algolia:**

**Email:**

[**stalingalaxy@gmail.com**](mailto:stalingalaxy@gmail.com)

**pwd:dlma123**

**App ID:**

**3WIBQ9UNH8**

**Secret id:**

**55b529a04492bff73bfc876c309e10f0**

**Brain tree:**

**Fname:stalin**

**Lname:Thomas**

**Vcompany:tvs**

## Create Sandbox account

**Fullname:stalin Thomas**

**Uname:**

**Stalin**

**Dilama**

## Sandbox Keys & Configuration

Merchant ID:

**4xmxz3nrxmpwv63n**

**Public Key:**

**v6gsqgg58wf2bbgn**

**Private\_key:**

**81af1a4c7f9362fb76ad6d920449e297**

Url:

$table->string('url')->unique();

Laravel-excel:

1. <https://github.com/Maatwebsite/Laravel-Excel>
2. <https://docs.laravel-excel.com/3.1/getting-started/>

**Laravel Pdf:**

<https://github.com/barryvdh/laravel-dompdf>

**composer require barryvdh/laravel-dompdf**

**After updating composer, add the ServiceProvider to the providers array in config/app.php**

**Barryvdh\DomPDF\ServiceProvider::class,**

You can optionally use the facade for shorter code. Add this to your facades or Alias code:

**'PDF' => Barryvdh\DomPDF\Facade::class,**

Create anew file named invoice.blade.php views/

Copy and paste bootstrap template

Web.php:

Route::get('/invoice', function () {

//return view('invoice');

$pdf = PDF::loadView('invoice');

return $pdf->download('invoice.pdf');

});

Run /invoice pdf created successfully

Next

**php artisan vendor:publish --provider="Barryvdh\DomPDF\ServiceProvider"**

domPDFfile created config/dompdf path

Page break:

**<style>**

**.page-break {**

**page-break-after: always;**

**}**

**</style>**

**<h1>Stalin</h1>**

**<div class="page-break"></div>**

**<h1>Thomas</h1>**

**Hint:**

**First page la stalin**

**Second page Thomas will display**

**Login with 'username' in laravel:**

<https://sujipthapa.co/blog/laravel-v55-login-register-with-username-or-email>

**Migrations table**

$table->string('username')->unique();

**User.php**

protected $fillable = [

'name', 'email', 'password', 'username',

];

**Register Controller**

return Validator::make($data, [

'username' => 'required|string|max:20|unique:users',

]);

protected function create(array $data)

{

return User::create([

'name' => $data['name'],

'username' => $data['username'],

'email' => $data['email'],

'password' => bcrypt($data['password']),

]);

}

**Register.blade.php**

<div class="form-group{{ $errors->has('username') ? ' has-error' : '' }}">

<label for="username" class="col-md-4 control-label">Username</label>

<div class="col-md-6">

<input id="username" type="text" class="form-control" name="username" value="{{ old('username') }}" required>

@if ($errors->has('username'))

<span class="help-block">

<strong>{{ $errors->first('username') }}</strong>

</span>

@endif

</div>

</div>

**Login Controller:**

use Illuminate\Http\Request;

protected function credentials(Request $request)

{

$field = filter\_var($request->get($this->username()), FILTER\_VALIDATE\_EMAIL)

? $this->username()

: 'username';

return [

$field => $request->get($this->username()),

'password' => $request->password,

];

}

Also, I changed few things in my login.blade.php, the email input field to be the **text** for username support.

<input **type="text"** class="form-control" name="email" placeholder="E-Mail / Username" value="{{ old('email') }}" required autofocus>

**Go to login.blade.php**

**Email type is text is set**

**RegisterController:**

return Validator::make($data, [

'name' => ['required', 'string', 'max:255'],

'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],

'profile\_image'=>['required'],

'username' => 'required|string|max:20|unique:users',

'password' => ['required', 'string', 'min:8', 'confirmed'],

]);

protected function create(array $data)

{

if(Input::hasFile('profile\_image')){

$file = Input::file('profile\_image');

$file->move(public\_path().'/uploads/',$file->getClientOriginalName());

$url = URL::to("/"). "/uploads/".$file->getClientOriginalName();

}

$data['profile\_image']= $url;

return User::create([

'name' => $data['name'],

'email' => $data['email'],

'username' => $data['username'],

'profile\_image'=>$data['profile\_image'],

'password' => Hash::make($data['password']),

]);

}

**Cashier Tutorial:**

**1.** composer require laravel/cashier

**2.** Schema::table('users', function ($table) {

$table->string('stripe\_id')->nullable()->collation('utf8mb4\_bin');

$table->string('card\_brand')->nullable();

$table->string('card\_last\_four', 4)->nullable();

$table->timestamp('trial\_ends\_at')->nullable();

});

Schema::create('subscriptions', function ($table) {

$table->increments('id');

$table->unsignedInteger('user\_id');

$table->string('name');

$table->string('stripe\_id')->collation('utf8mb4\_bin');

$table->string('stripe\_plan');

$table->integer('quantity');

$table->timestamp('trial\_ends\_at')->nullable();

$table->timestamp('ends\_at')->nullable();

$table->timestamps();

});

**Php artisan migrate**

use Laravel\Cashier\Billable;

class User extends Authenticatable

{

use Billable;

}

**Go to .env**

**STRIPE\_KEY = pk\_test\_t9JIYDUJDlhjoUBwKWyH5sr900NFtlF5bj**

**STRIPE\_SECRET = sk\_test\_xlJ1nlMXtUESup7CzmDk3Rw70079C3Hfuj**

create file stripe.blade.php

<https://stripe.com/docs/stripe-js>

**javascript added just before body element**

<**script** *src*="https://js.stripe.com/v3/"></**script**>

**Js code code**

**</body>**

**Make one controller**

public function index(){

return view('stripe');

}

Route::get('/charge', 'ChargeController@index');

Route::post('/charge', 'ChargeController@store');

Add

<form action="{{ url('/charge') }}" method="post" id="payment-form">

{!! csrf\_field() !!}

Run

/charge

Visa card 4242 4242 4242 4242 zip:54000

public function store(Request $request){

dd($request->all());

exit;

}

array:2 [▼

"\_token" => "**4Onbfb7GK3q1FtovUgwvqscF7aygzvNN8YCvowbx**"

"stripeToken" => "**tok\_1EMFAbBYaJeYRTvcvWB0YpyR**"

]

$user = User::find(1);

$user->newSubscription('main', 'premium')->create($token);

$stripeToken = $request->get('stripeToken');

$user = User::find(1);

$plan = “monthly”;

$user->newSubscription('main',$plan)->create($token);

Go to stripe panel;

Click product tab

Create new product->Enter name ->create product btn

Add pricing plan

Nick name = monthly

Price per unit:9 dollar

Monthy select dropdown

Save

Go to url run it

Go to stripe’s subscription tab

Link display click that link

public function store(Request $request){

dd($request->all());

exit;//

// return view('stripe');

$stripeToken = $request->get('stripeToken');

$plan = "plan\_EpuOvjJRE1zIbq";

$user = User::find(1);

$user->newSubscription('main', $plan)->create( $stripeToken);

}

mani@qpayindia.com

Go to table user and subscription table

Custid is added in usertable in stripe id

public function index(){

$user = User::find(1);

$charge=$user->charge(9000);

// dd($charge);

return view('stripe');

}

**Middleware auth:**

<http://localhost:8000/admin/>

Route::group(['prefix' => 'admin'],function()

{

Route::get('/', function(){

return "admin login/dashboard page";

});

Route::get('/departments', function(){

return "departments page";

});

Route::get('/employees', function(){

return "employees page";

});

});

Route::group(['prefix' => 'admin','middleware' => 'auth'],function()

{

Route::get('/', function(){

return "I am Admin Page";

});

Route::get('/departments', function(){

return "departments page";

});

Route::get('/employees', function(){

return "employees page";

});

});